

Problem statement-8 Hibernate

```
package com;
import java.math.BigDecimal;
import java.util.Date;
public class EProductm {
    private long ID;
    private String name;
    private BigDecimal price;
    private Date dateAdded;
    private ProductParts parts;

    public EProductm() {

    }

    public long getID() {return this.ID; }
    public String getName() { return this.name;}
    public BigDecimal getPrice() { return this.price;}
    public Date getDateAdded() { return this.dateAdded;}
    public ProductParts getParts() { return this.parts;}

    public void setID(long id) { this.ID = id;}
    public void setName(String name) { this.name = name;}
    public void setPrice(BigDecimal price) { this.price = price;}
    public void setDateAdded(Date date) { this.dateAdded = date;}
    public void setParts(ProductParts parts) { this.parts = parts;}
}
```

```
package com;
public class ProductParts {
    private String hdd;
    private String cpu;
    private String ram;

    public String getHdd() { return this.hdd;}
    public String getCpu() { return this.cpu;}
    public String getRam() { return this.ram;}

    public void setHdd(String value) { this.hdd= value;}
    public void setCpu(String value) { this.cpu= value;}
    public void setRam(String value) { this.ram= value;}
}
```

<!DOCTYPE hibernate-mapping PUBLIC

```

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com">
  <class name="EProductm" table="eproduct">
    <id name="ID" type="long" column="ID">
      <generator class="identity"/>
    </id>
    <property name="name" type="string" column="NAME"/>
    <property name="price" type="big_decimal" column="PRICE"/>
    <property name="dateAdded" type="timestamp" column="DATE_ADDED"/>
    <component name="parts" class="com.ProductParts">
      <property name="hdd" column="parts_hdd" type="string"
/>
      <property name="cpu" column="parts_cpu" type="string"
/>
      <property name="ram" column="parts_ram" type="string"
/>
    </component>
  </class>
</hibernate-mapping>

```

```

<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <!-- Database connection settings -->
    <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="connection.url">jdbc:mysql://localhost:3306/ec</property>
    <property name="connection.username">root</property>
    <property name="connection.password">Qwer123!@#</property>
    <mapping resource="com.EProductm.hbm.xml"/>
  </session-factory>
</hibernate-configuration>

```

```

package com;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;

```

```

import javax.servlet.http.HttpServletResponse;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
public class ProductDetailsMapping extends HttpServlet {
    private static final long serialVersionUID = 1L;
    // public ProductDetailsMapping() {
    //     super();
    //     // TODO Auto-generated constructor stub
    // }
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        try {
            SessionFactory factory = HibernateUtil.getSessionFactory();

            Session session = factory.openSession();

            List<EProduct> list = session.createQuery("from EProductm").list();

            PrintWriter out = response.getWriter();
            out.println("<html><body>");

            out.println("<b>Component Mapping</b><br>");
            for(EProduct p: list) {
                out.println("ID: " + String.valueOf(p.getID()) + ", Name:
" + p.getName() +
                    ", Price: " + String.valueOf(p.getPrice())
+ ", Date Added: " + p.getDateAdded().toString());
                ProductParts parts = p.getParts();
                out.println("Parts =" + parts.getCpu() + ", " +
parts.getHdd() + ", " + parts.getRam());
                out.println("<hr>");
            }

            session.close();
            out.println("</body></html>");

        } catch (Exception ex) {
            throw ex;
        }
    }
    // protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    //     // TODO Auto-generated method stub
    //     doGet(request, response);
    // }
}

```

